* **[Auth sponsors]**
  + La ruta POST /api/economy/transfer valida que el fromUserId sea patrocinador. Si más adelante abrimos esto fuera del dashboard admin, conviene añadir autenticación por usuario (JWT/session/Telegram login) para evitar usos indebidos.
* **[Optimización de listado]**
  + listUsersWithFires() consulta SISMEMBER por usuario. Para lotes más grandes, podemos optimizar leyendo todo el set de patrocinadores (SMEMBERS) y usando un Set en memoria para chequear membresía localmente.

Actua como un profesional organizando proyectos investiga en la web y vamos a replantear lo que tenemos actualmente y vas a tomar en cuenta a continuacion:  
  
una vez que terminemosde remover el dominó nos vamos a centrar en la estética y pulir todos los detalles que hay solo con 2 juegos  
  
primero, todos los juegos tienen modalidad "amistoso" (no requiere fuego)  
para permitir que todos puedan participar,  
tambien tienen el modo fuego, requiere fuego para poder comenzar la partida, si el host abre una sala,amistosa, para que entre la gente y luego de que esten en la sala cambia el modo a fuego, no va a permitir iniciar la partida si alguien no tiene fuegos y les va a mostrar a todos los usuarios con un aura roja al rededor del usuario indicando que ese usuario no tiene fuego, asi manejaremos ese error, tomar en cuenta que todo va ir desarrollandose con estas intenciones para permitir interaccion con la plataforma sin obligar al usuario que disponga fuegos para jugar de manera amistosa  
  
vamos a organizar la data, primero, vamos a analizar lo que NO estamos usando y lo eliminamos por ejemplo ngrok, igualmente revisaremos.  
  
ya estamos definiendo, tenemos una organizacion asi  
  
tenemos una simulacion de token (fuego)  
tenemos una plataforma de minijuegos  
en la plataforma de minijuegos se le da uso al token  
en la plataforma de juegos se le agregan juegos como: tic tac toe, bingo, etc.  
la plataforma debe ser modular y permitir que un juego sea agregado o removido facilmente, igualmente la estructura del token debe acoplarse facilmente a la incorporacion de nuevos proyectos ya que seguiremos ampliando  
el token debe estar separado para permitir su futura implementacion en una blockchain.  
  
ya que tenemos un supply mi intencion es que al incorporar una tienda, esta tienda tendrá una dashboard propia para gestionar sus ventas, compras, articulos, inventarios, estadisticas, informes y cada vez que un cliente realice una compra en la tienda se acuña un nuevo token del supply maximo, y debe aparecer en nuestro dashboard las monedas acuñadas por cada tienda que comienzan a entrar en circulacion. desde nuestro dashboard debo poder cambiar la cantidad de monedas que se pueden acuñar por cada tienda.  
  
tenemos que desarrollar un sistema de quema de tokens,de este modo seguiremos en la estructura que tenemos  
  
Los usuarios pueden canjear la cantidad de tokens que deseen de los que tengan disponibles, este canje será quemarlos para reducir el max-supply, tambien podran "comprar" algunas cosas y esta compra con tokens serà una quema directa de estos tokens asi que se debe implementar un modo de NFT de tal modo yo pueda agregarlos desdes el dashboard y aparezcan en otra interfaz que un usuario pueda entrar en un "minijuego" de articulos y canjear sus tokens por ese "NFT".